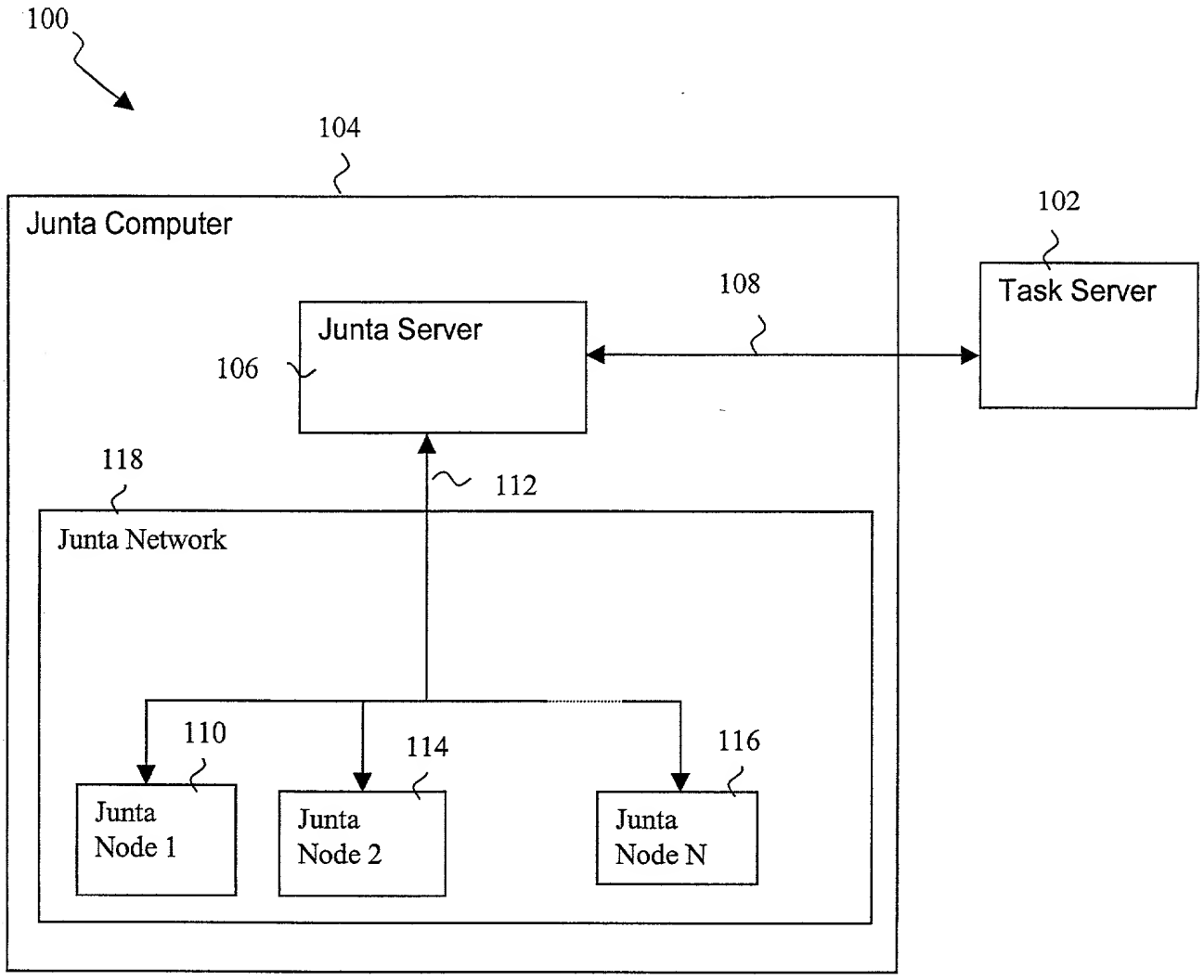
100
104
Junta Computer
102
Task Server
108
Junta Server
106
118
112
Junta Network
110
114
116
Junta
Node 1
Junta
Node 2
Junta
Node N

Figure 1

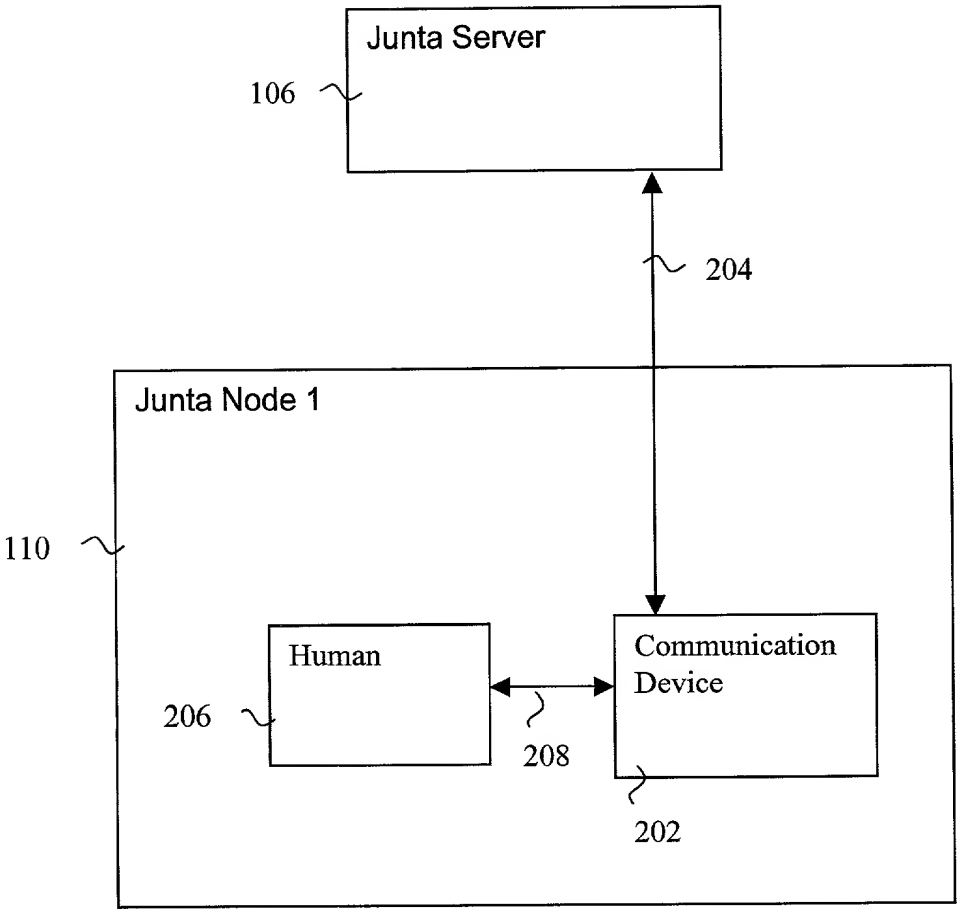
Junta Server
106
204
Junta Node 1
110
Human
206
208
Communication Device
202

Figure 2

Figure 3a

302
```
/* Defines the various types of input   */
typedef enum {
        TEXT   = 0x1,
        SPEECH = 0x2,
} junta_input_domains_t;
```

304
```
/* Defines the various output types   */
typedef enum {
        TEXT   = 0x1,
        NUMBERS = 0x2,
        TUPLES = 0x4,
} junta_output_domains_t;
```

306
```
/* A function maps an input to an output */
typedef struct {
        junta_input_domains_t input;
        junta_output_domains_t output;
} junta_function_s;
```

308
```
/* Create a new problem on the Junta Computer */
junta_return_t junta_new_problem(junta_function_s function_type,
                                 junta_id *problem_id);
```

310
```
/* The various attributes we might want to set on a problem */
typedef enum {
        ACCURACY,
        SECURITY,
        TIMEOUT,
        MAXIMUM_TIME,
        COST_PER_TASK,
        MAXIMUM_COST,
} junta_attribute_type_t;
```

312
```
/* The accuracy functions we have available right now */
typedef enum {
        MAJORITY_WINS,
        SPECIFIC_ACCURACY,
        AT_LEAST_N,
} junta_accuracy_functions_t;
```

314
```
/* The security levels we export for now */
typedef enum {
        STRICT,
        LAX,
        NONE,
} junta_security_levels_t;
```

Figure 3b

316
```
/* Specify the parameters for accuracy */
typedef struct {
    junta_accuracy_functions_t function_type;
    union {
        struct {
            int num_voters;
        } majority_wins;
        struct {
            float probability;
            float confidence;
        } specific_accuracy;
        struct {
            int n;
        } at_least_n;
    } u;
} junta_accuracy_s;
```

318
```
/* Specify the security parameters */
typedef struct {
    junta_security_levels_t security_level;
} junta_security_s;
```

320
```
/* All the attributes */
typedef union {
    junta_accuracy_s accuracy;
    junta_security_s security;
    int timeout;
    int maximum_time;
    float cost_per_task;
    float max_cost;
} junta_attribute_s;
```

322
```
/* Set some attributes on it */
junta_return_t junta_set_attributes(junta_id problem_id,
                                    junta_attribute_type_t attribute_type,
                                    junta_attribute_s attributes);
```

324
```
typedef union {
    char **text;
    void **speech_data;
} junta_input_data_s;
```

326
```
typedef union {
    int *numbers;
    char *text;
} junta_output_data_s;
```

328
```
/* Supply the input values */
junta_return_t junta_input(junta_id problem_id,
                           int input_id,
                           junta_input_data_s input);
```

330
```
/* Set the allowed range of output values */
junta_return_t junta_set_output_values(junta_id problem_id,
                                       junta_output_data_s output);
```

332
```
/* Get the output for the input set specified. */
junta_return_t junta_get_output(junta_id problem_id,
                                int input_id,
                                junta_output_data_s output);
```

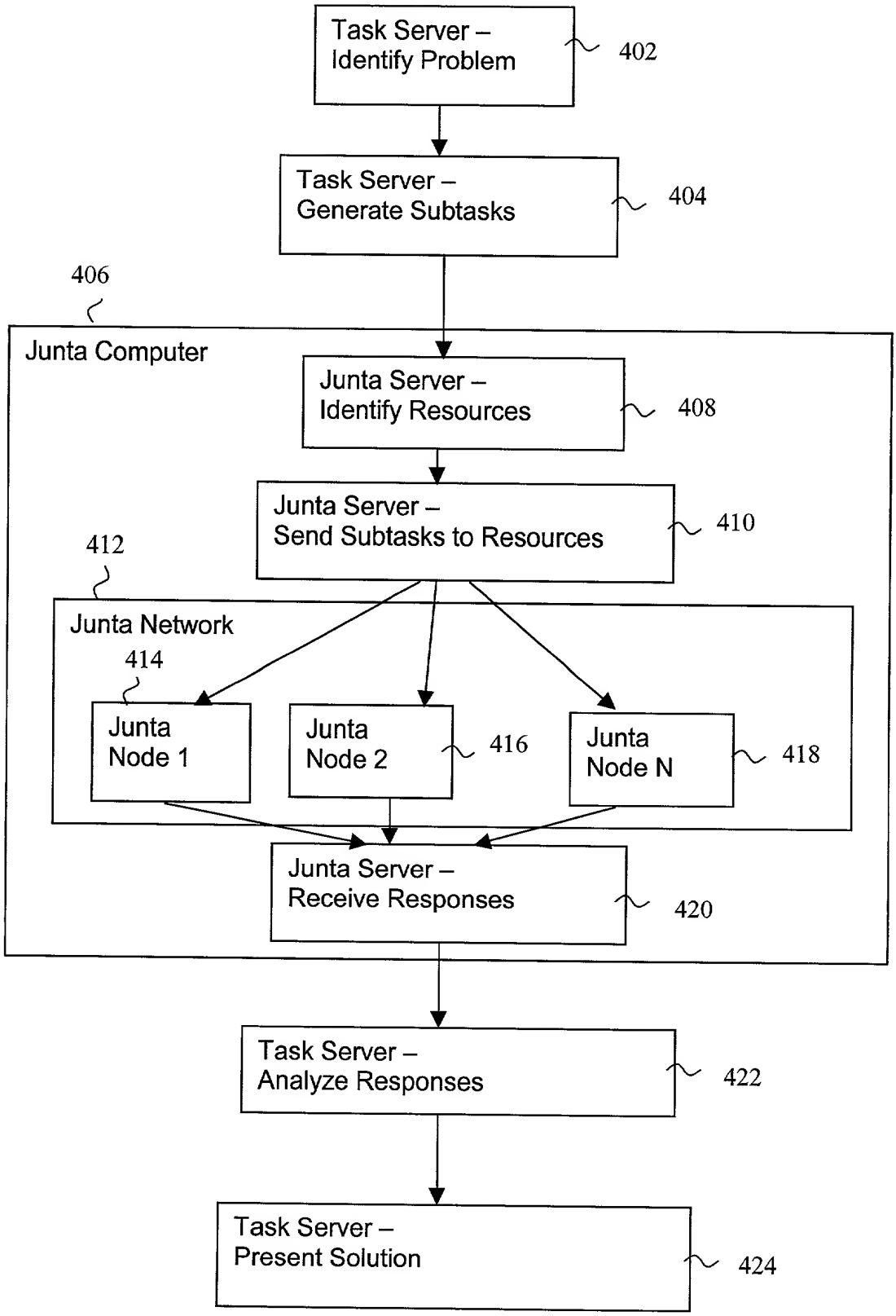
Task Server – Identify Problem
402
Task Server – Generate Subtasks
404
406
Junta Computer
Junta Server – Identify Resources
408
Junta Server – Send Subtasks to Resources
410
412
Junta Network
414
Junta Node 1
Junta Node 2
416
Junta Node N
418
Junta Server – Receive Responses
420
Task Server – Analyze Responses
422
Task Server – Present Solution
424

Figure 4